

AD-A125 029

A COMPUTER PACKAGE FOR THE COMPOSITE CRITERION MODEL

1/1

(U) GEORGIA INST OF TECH ATLANTA PRODUCTION AND
DISTRIBUTION RESEARCH CENTER L F MCGINNIS ET AL.

UNCLASSIFIED

JAN 81 PDRC-82-13 N00014-79-C-0779

F/G 12/1

NL

END

DATE

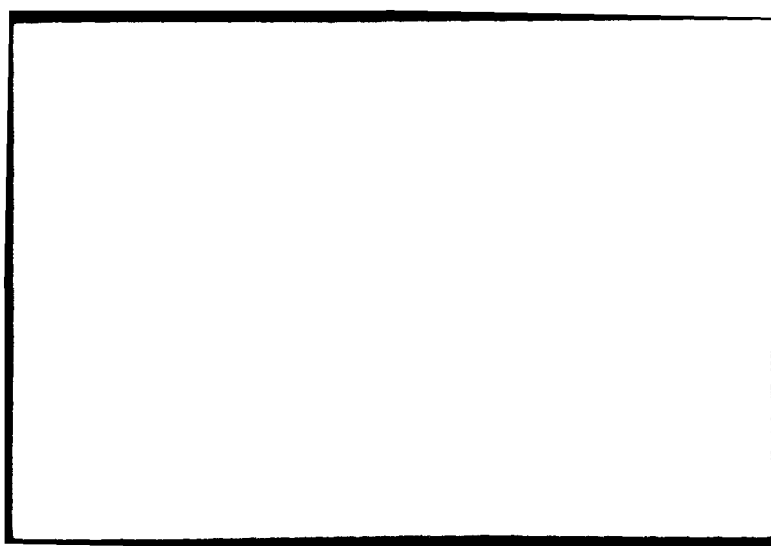
FILED

1-83

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A



A COMPUTER PACKAGE FOR THE
COMPOSITE CRITERION MODEL

by

L. F. McGinnis[†]

R. K. Runyan

PDRC-82-13

[†]School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332
1-404-894-2363

DTIC
ELECTE
FEB 28 1983
S B

This research was supported by ONR Research Contract N00014-79-C-0779 through the University of Massachusetts at Amherst, and ONR Research Contract N00014-80-k-0709. Reproduction in whole or in part is permitted for any purpose of the U.S. Government.

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

- a -
Abstract

This report describes a general purpose computer package for analysis of the composite criterion model of Srinivasan and Shocker. (Psychometrika- Vol. 38, No. 4, p. 473). A standard format allows data for a wide range of applications to be analyzed using this package. The report describes the development of the package, including the data base and the FORTRAN codes for performing the analysis. The sample problem used by Srinivasan and Shocker is analyzed to demonstrate the package.

This model was proposed as a technique for analyzing a decision maker's preferences when they are multiple conflicting criteria.



Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
PER LETTER	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	

A COMPUTER PACKAGE FOR THE COMPOSITE CRITERION MODEL

The composite criterion model was proposed by Srinivasan and Shocker [4] as a technique for analyzing a decision maker's preferences when there are multiple conflicting criteria. The model is based on the assumed existence of a value function which assigns a cardinal score to any alternative. Associated with alternative i is a vector $[a_{ij}]$ of attribute values. The value function is assumed to be:

$$s_i = v([a_{ij}]) = \sum_j w_j a_{ij} \quad (1)$$

The goal of the composite criterion model is to use a limited number of explicit pairwise comparisons to reveal the proper values of the weights, w_j , for a given decision maker.

Given two alternatives, i and j , if the decision maker prefers i over j , then it is assumed that $s_i \geq s_j$, i.e.,

$$\sum_k w_k a_{ik} - \sum_k w_k a_{jk} \geq 0$$

$$\text{or} \quad \sum_k (a_{ik} - a_{jk}) w_k \geq 0 \quad (2)$$

Thus, if we have a set, Ω , of pairwise comparisons, there is a corresponding set of constraints, (2), which must be satisfied by the weights w_k .

Since it is possible that the decision maker may err in the preference responses (or that the linear value function assumption is not completely satisfied), the feasible set defined by (2) for $(i, j) \in \Omega$ may be empty. In that case, we would like to determine a "best" set of weights.

The composite criterion model, as proposed by Srinivasan and Shocker, determines the set of weights which minimizes the sum of the infeasibilities

in the constraints (2):

$$\text{minimize} \quad \sum_{(i,j) \in \Omega} z_{ij} \quad (3)$$

$$\text{s.t.} \quad \sum_{k \in P} a_{ijk} w_k + z_{ij} \geq 0 \quad (i,j) \in \Omega \quad (4)$$

$$\sum_{k \in P} \sum_{(i,j) \in \Omega} a_{ijk} w_k = 1 \quad (5)$$

$$z_{ij} \geq 0 \quad (i,j) \in \Omega \quad (6)$$

$$w_k \geq 0 \quad k \in P_1 \quad (7)$$

$$w_k \leq 0 \quad k \in P_2 \quad (8)$$

$$w_k \text{ urs} \quad k \in P_3 \quad (9)$$

$$P = P_1 \cup P_2 \cup P_3; \quad P_i \cap P_j = \emptyset \quad i \neq j$$

Note that in order to determine the weights, w_k , the linear programming problem (3) - (9) must be solved.

Any composite criterion problem is defined by the sets Ω , P_1 , P_2 , P_3 and the vectors $[a_{ijk}]$. Thus, it is reasonable to develop a general purpose data base and solution procedure. The package we have developed is based on the overall design illustrated in Figure 1. The LP optimizer is the XMP suite of FORTRAN routines developed by Marsten [1]. The entire package is written in FORTRAN (ANSI 66), is reasonably portable, and can accomodate problems with as many as 30 restricted attributes (or 15 unrestricted attributes) and 100 pairwise comparisons.

Parameter sizes on the CCMODL program can be increased to accommodate

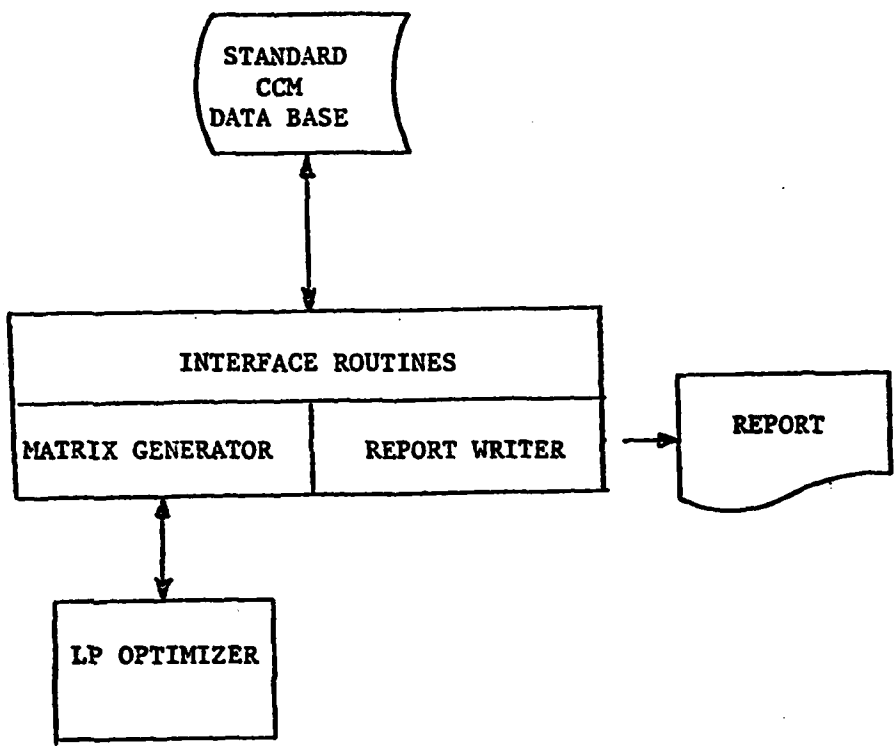


Figure 1. Overall System Design

any size problem with the amount of array space increasing as a function of the model parameters. Let:

MAXFPC = maximum number of forced-paired comparisons,

MAXWTS = maximum number of weights that can be specified where every unrestricted weight counts as two weights,

A = number of words required for array storage,

Then:

$$A \geq 63 \text{ MAXFPC} + 7 \text{ MAXWTS} + 4 (\text{MAXFPC}) (\text{MAXWTS}) + \frac{12 (\text{MAXFPC} + 1) (\text{MAXWTS}) (\text{MAXFPC})}{\text{MAXWTS} + 3 \text{ MAXFPC} + 1} + 151$$

STANDARD CCM DATA BASE

The data base contains a complete description of the problem, from which the interface routines generate the LP coefficients. There are five record groups in the data base:

- group 1: this is a title record to allow a description to appear on the report; (10A8) format,
- group 2: this gives the magnitude of the problem; (KATTA, KSTIMLI, KFCOMP), where,
 KATTA = number of attributes
 KSTIMLI = number of stimuli or alternatives
 KFCOMP = number of paired comparisons
 (3I5) format
- group 3: a set of flags to indicate sign restrictions on the weights; the first record is

$$\text{KWFLAG} = \begin{cases} 1 & \text{if } P = P_1 \\ -1 & \text{if } P = P_2 \\ 2 & \text{if } P = P_3 \\ 0 & \text{otherwise} \end{cases}$$

(I2) format;

If KWFLAG = 0, then KATTA flags KW(i) are required, one for each attribute

$$\text{KW}(i) = \begin{cases} 1 & \text{if } i \in P_1 \\ -1 & \text{if } i \in P_2 \\ 2 & \text{if } i \in P_3 \end{cases}$$

(I2) format.

group 4: the set Ω ; KFCOMP pairs (i, j) are required where i is preferred to j (2I4) format

group 5: attribute values; for each stimulus, a set of KATTA values are required, in (8F10.0) format; KSTMLI sets are required.

Example. Srinivasan and Shocker describe a small problem with five stimuli and three attributes. The data for this problem is given below:

$\Omega = \{(1,2), (3,1), (1,4), (1,5), (2,3), (2,4), (5,2), (3,4), (3,5), (5,4)\}$

* $P = P_3$

<u>Stimulus</u>	<u>Attribute Vector</u>
1	(2,6,6)
2	(3,4,7)
3	(5,4,4)
4	(6,3,2)
5	(8,1,3)

The standard CCM data set for this problem is given in Table 1.

* There is a small difference between this problem and the Srinivasan and Shocker problem in [4], the difference being that in [4] all weights are restricted to be non-negative while in this problem all weights are unrestricted.

LP OPTIMIZER

The LP optimizer is the XMP package developed by Marsten [1]. XMP implements a revised simplex algorithm and uses the LA05 subroutines to manage an LU factorization of the basis matrix from the Harwell library. XMP is written in a "Vanilla" FORTRAN where only very simple features of the language have been used. XMP operates in main memory where the memory space that XMP uses must be provided by the calling program. XMP is hierarchically structured and any XMP subroutine may be called at any level of the hierarchy.

The design objectives of XMP were flexibility, extendability, and reliability with execution time as a secondary objective. XMP contains 46 subroutines, 10500 lines of FORTRAN and requires 10395 words for the object code.

INTERFACE ROUTINES

The interface code MCCMDL, consists of 551 lines of FORTRAN and is internally documented. The code requires 1246 words for the object code and 29708 words for arrays (this includes the arrays required for XMP). As presently configured, MCCMDL can accommodate 100 stimuli, 30 attributes and 100 paired comparisons. The number of paired comparisons is the critical parameter, since it increases both the number of variables and the number of constraints in the LP problem.

The CCMODL program was used to solve the problem described in the data set listed in Table 1, the output is shown in Table 2. The results shown in Table 2 are the same results obtained by Srinivasan and Shocker in [4]. The run was completed in .486 CP seconds.

As discussed in [4] and as seen in Table 2, stimuli 1, 2, 5 have the same score. This occurred because of the number of intransitivities in the pairwise comparisons in the example. Since every Z_i except $Z_5 = .02174$, is equal to zero this implies that every preferred stimulus in a pairwise comparison has a higher score than the other stimuli in the comparison except in comparison 5. In comparison 5 the Decision Maker preferred stimulus 2 to stimulus 3 when comparing scores we see that stimulus 3 has a higher score (by .02174) than does stimulus 2.

A listing of the FORTRAN source for MCCMDL is given in Appendix A.

REFERENCES

1. Marsten, Roy E. (1979), "XMP: A Structured Library of Subroutines for Experimental Mathematical Programming," Technical Report No. 351, Management Information Systems, The University of Arizona, Tucson, Arizona 85721.
2. Parker, B. R., and Srinivasan, V. (1976), "Consumer Preference Approach to Planning of Rural Primary Healthcare Facilities," Operations Research, Vol. 24 (5), pp. 991-1025.
3. Srinivasan, V. (1976), "Linear-Programming Computational Procedures for Ordinal Regression," Journal of the Association for Computing Machinery, Vol. 23 (3), pp. 475-487.
4. Srinivasan, V., and Shocker, A. D. (1973), "Estimating the Weights for Multiple Attributes on a Composite Criterion Using Pairwise Judgment," Psychometrika, Vol. 38 (4), pp. 473-492.

Table 1. Standard CCNODL Data Set for Srinivasan-Shocker Example

Trial 1 For The C.C. Model Program, All Weights Are Unrestricted

	3	5	10
2			
1	2		
3	1		
1	4		
1	5		
2	3		
2	4		
5	2		
3	4		
3	5		
5	4		
	2.	6.	6.
	3.	4.	7.
	5.	4.	4.
	6.	3.	2.
	8.	1.	3.

Table 2. Result for Srinivasan-Shocker Example

TRIAL 1 FOR THE C.C. MODEL PROGRAM, ALL WEIGHTS ARE UNRESTRICTED

THE OPTIMAL OBJECTIVE FUNCTION VALUE IS Z= .02174

WEIGHT#	***VALUE**
1	.23913
2	.19565
3	.15217

Z	***VALUE**
1	0.00000
2	0.00000
3	0.00000
4	0.00000
5	.02174
6	0.00000
7	0.00000
8	0.00000
9	0.00000
10	0.00000

STIMULUS	***SCORE**
1	2.56522
2	2.56522
3	2.56696
4	2.32609
5	2.56522

APPENDIX A

Source Listing for MCCMDL

PROGRAM MCCMDL(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE9)

THE PURPOSE OF THIS PROGRAM IS TO SOLVE THE COMPOSITE CRITERION MODEL
BY THE USE OF LINEAR PROGRAMMING. THE L.P. MODEL WILL BE SOLVED BY THE
XMP SUBROUTINE PACKAGE.

SINCE THE XMP PACKAGE OF SUBROUTINES IS SO WELL DOCUMENTED THE COMMENTS
IN THIS PROGRAM WILL ONLY BE DIRECTED TOWARD THE VARIABLES AND LOGIC OF
THIS PROGRAM ONLY. THE PARTS OF THIS PROGRAM COVERED IN THE XMP DOCUMENT-
ATION WILL BE SET OF BY THE FOLLOWING -----.

THIS PROGRAM IS WRITTEN BY RONALD K. RUNYAN UNDER THE DIRECTION OF
DR. LEON F. MCGINNIS.

GEGORIA INSTITUTE OF TECHNOLOGY, FALL 1960.

DIMENSION STATEMENTS

DIMENSION STIML1(100,30),KW(30),WEIGHT(30),RHALF(101,30),Z2(100)
DIMENSION HEADER(10),KWDEF(30)
INTEGER OMEGA(100,2)

FOR CHANGES IN THE PARAMETERS,THE ARRAY SIZES SHOULD BE THE
FOLLOWING SIZE.

STIML1(MAXFPC,MAXWTS)
KW(MAXWTS)
WEIGHT(MAXWTS)
RHALF(MAXFPC+1,MAXWTS)
Z2(MAXFPC)
HEADER(10)
OMEGA(MAXFPC,2)
KWDEF(MAXWTS)

----- DIMENSION AND VARIABLE DEFINITION FOR XMP

REAL B(101),BASCB(101),BASLB(101),BASUB(101),
X BETAR(101),BOUND,CANDA(101,5),CANDCJ(5),CJ,COLA(101),
X LJ,MEMORY(20700),UJ,UZERO(101),XBZERO(101),YQ(101),Z

INTEGER BNDTYP,COLLEN,COLMAX,DFEASQ,DTERM,DUNBR,
X FACTOR,IOERR,ILOG,ITER,
X ITER1,ITER2,LENMA,LENMI,LENNY,LOOK,M,MAPA(20),
X MAPI(20),MAXA,MAXM,MAXN,N,NTYPE2,P,PRINT,TERMIN,
X UNBDDQ,BASIS(101),CAND(5),CANDI(101,5),CANDL(5),
X COLI(101),ROWTYP(101),STATUS(331)

REAL VALUE

INTEGER IOIN,NCOLSA,TT,SS
COMMON STIML1,KW,WEIGHT,RHALF,Z2,HEADER,OMEGA,B,BASCB,BASLB
-,BASUB,BETAR,CANDA,CANDCJ,COLA,MEMORY,UZERO,XBZERO,YQ,MAPA,
-MAPI,BASIS,CAND,CANDI,CANDL,COLI,ROWTYP,STATUS,KWDEF

```

C WITH CHANGES IN THE BELOW(XMP) PARAMETERS THE SIZE OF THE PROBLEM
C IS LIMITED ONLY BY THE AVAILABILITY OF COMPUTER CORE.
C FOR PROBLEMS LARGER THE 30 RESTRICTED ATTRIBUTES(15 UNRESTRICTED
C ATTRIBUTES) AND 100 PAIRWISE COMPARISONS SET THE BELOW PARAMETERS
C TO THE FOLLOWING VALUES.
C MAXA=MAXWTS*MAXFPC+MAXM
C MAXM=MAXFPC+1
C MAXN=MAXWTS+3*MAXFPC+1
C COLMAX=MAXM
C LOOK=MAXM
C LERNY>=11*MAXM+2*MAXN+2*MAXA+0+3*(4*DENSE*MAXM**2)
C DENSE=MAXA/(MAXM*MAXN)
C
C
C
C

```

INITIALIZE VARIABLES FOR XMP

```

C IOIN=5
C IOERR=9
C IOLOG=9
C MAXA=3331
C MAXM=101
C MAXN=331
C COLMAX=101
C P=5
C LOOK=331
C FACTOR=10
C LERNY=20700
C BNDTYP=1
C PRINT=0

```

```

C
C THE OUTPUT WRITTEN FOR THE PRINT STATEMENTS WITHIN XMP WILL BE
C PLACED IN A FILE CALLED TAPE9, THIS FILE WILL BE A LOCAL FILE
C WHEN THIS PROGRAM IS RUN FROM A TERMINAL. TO READ THE CONTENTS
C OF THIS FILE YOU DO THE FOLLOWING AFTER THE PROGRAM HAS FINISHED
C RUNNING.
C REWIND,TAPE9
C COPY,TAPE9
C THIS FILE IS LOST IF NOT "SAVE"-D AFTER THE RUN.
C
C
C
C

```

INITIALIZE ALL PARAMETERS OF THE MODEL

```

C MAXATT=30
C MAXSTI=100
C MAXFPC=100
C MAXWTS=30
C ZERO OUT THE KWDEF AND THE WEIGHT ARRAYS
C DO 7 KARRAY=1,MAXATT
C KWDEF(KARRAY)=0
C WEIGHT(KARRAY)=0.
C 7 CONTINUE

```

VARIABLE LIST


```

1206 FORMAT(1H ,////,21X,"WRITTEN BY: R. K. RUMYAN",//,
-21X,"FACULTY SPONSOR: L. F. MCGILVERIS",//,
-21X,"GEORGIA INSTITUTE OF TECHNOLOGY",//,
-21X,"VERSION 1",//,21X,"NOVEMBER 15, 1960 ",//)
WRITE(6,215)
C READ IN THE HEADER OR PROBLEM STATEMENT OF THE PROGRAM
READ(5,1740) (HEADER(IHEAD),IHEAD=1,10)
1740 FORMAT(10A6)
READ(5,10) KATTA,KSTMLI,KFCOMP
10 FORMAT(3I5)
C CHECK FOR ZERO VALUES OF KATTA,KSTMLI,KFCOMP
IF(KATTA.LE.0) GO TO 44
IF(KSTMLI.LE.0) GO TO 44
IF(KFCOMP.LE.0) GO TO 44
GO TO 77
44 WRITE(6,80)
88 FORMAT(1H0," CHECK THE VARIABLES,KATTA,KSTMLI,KFCOMP, FOR VALUES
-LESS THAN OR EQUAL TO 0. *ERROR 9")
KERROR=1
GO TO 9999
77 CONTINUE
NUMWTS=KATTA
READ(5,15) KWFLAG,(KW(IST),IST=1,KATTA)
15 FORMAT(32I2)
IF(KWFLAG.EQ.0) GO TO 1682
C INITIALIZE THE VALUES OF EACH KW(I) IF KWFLAG IS NOT EQUAL TO 0.
DO 1681 IST=1,KATTA
KW(IST)=KWFLAG
1681 CONTINUE
1682 CONTINUE
C
C ERROR MESSAGES
C *****
C *
C * ERROR 1 MORE THAN MAXATT ATTRIBUTES HAVE BEEN SPECIFIED *
C * ERROR 2 MORE THAN MAXSTI STIMULI HAVE BEEN SPECIFIED *
C * ERROR 3 MORE THAN MAXFPC FORCE PAIRED COMPARISONS HAVE BEEN SPECIFIED*
C * ERROR 4 INVALID VALUE OF KWFLAG HAS BEEN SPECIFIED *
C * ERROR 5 NUMWTS IS GREATER THAN MAXWTS *
C * ERROR 6 A SPECIFIED WEIGHT ,I.E. KW(I), HAS AN INVALID VALUE *
C * ERROR 8 A STIMULUS IS OUT OF RANGE FOR A FORCED COMPARISON *
C * ERROR 9 EITHER KATTA,STIMLI,OR KFCOMP IS <=0 *
C *****
C
C BEGIN CHECKING FOR OTHER ERRORS
C
KERROR=0
IF(KATTA.LE.MAXATT) GO TO 20
WRITE(6,30) KATTA,MAXATT
30 FORMAT(1H0,16," ATTRIBUTES HAVE BEEN SPECIFIED, THIS IS IN EXCESS
- OF THE",15," ATTRIBUTES THAT CAN BE SPECIFIED. *ERROR 1")
KERROR=1
20 IF(KSTMLI.LE.MAXSTI) GO TO 35

```

```

WRITE(6,45) KSTMLI,MAXSTI
45 FORMAT(1H0,I6," STIMLI HAVE BEEN SPECIFIED, THIS IS IN EXCESS OF
-THE",I6," STIMLI THAT CAN BE SPECIFIED. *ERROR 2")
KERROR=1
35 IF(KFCOMP.LE.MAXFPC) GO TO 50
WRITE(6,60) KFCOMP,MAXFPC
60 FORMAT(1H0,I6," FORCED PAIRED COMPARISONS HAVE BEEN SPECIFIED, THI
-S IS IN EXCESS OF THE ALLOWABLE",I6,". *ERROR 3")
KERROR=1
50 IF(KWFLAG.EQ.-1) GO TO 52
IF(KWFLAG.EQ.0) GO TO 52
IF(KWFLAG.EQ.1) GO TO 52
IF(KWFLAG.EQ.2) GO TO 52
WRITE(6,110) KWFLAG
110 FORMAT(1H0," AN INVALID VALUE OF KWFLAG=",I4," HAS BEEN SPECIFIED.
- *ERROR 4")
KERROR=1
52 CONTINUE
IF(KERROR.EQ.1) GO TO 9999
IF(KWFLAG.NE.0) GO TO 1267
DO 2231 JUMP=1,KATTA
KLZ=KW(JUMP)
IF(KLZ.EQ.-1) GO TO 2231
IF(KLZ.EQ.2) GO TO 2231
IF(KLZ.EQ.1) GO TO 2231
WRITE(6,5431) JUMP,KLZ
5431 FORMAT(1H,"AN INVALID VALUE FOR A SPECIFIED "
-,"WEIGHT DIRECTION HAS BEEN ENCOUNTERED",/
-," CHECK THE",I4," ENTRY FOR A VALUE OF",I3," *ERROR 6")
KERROR=1
2231 CONTINUE
1267 CONTINUE
C CHECK TO SEE IF THE LIMIT MAXWTS HAS BEEN EXCEEDED
IF(ABS(KWFLAG)-1) 65,120,75
C WHEN ALL WEIGHTS ARE UNRESTRICTED
75 L22=MAXWTS/2
IF(KATTA.LE.L22) GO TO 120
GO TO 85
65 KKTEMP=0
DO 90 ITT=1,NUMWTS
KKTEMP=KKTEMP+ABS(KW(ITT))
90 CONTINUE
IF(KKTEMP.LE.MAXWTS) GO TO 120
C THE LIMIT MAXWTS HAS BEEN EXCEEDED
85 WRITE(6,105)
105 FORMAT(1H0," A MAXIMUM OF 30 WEIGHTS CAN BE SPECIFIED WHERE EVERY
-UNRESTRICTED WEIGHT COUNTS AS 2."/ " THIS 30 WEIGHT LIMIT HAS BEEN
- EXCEEDED. *ERROR 5")
KERROR=1
120 IF(KERROR.EQ.1) GO TO 9999
C READ IN THE SET OF FORCED-PAIRED COMPARISONS(OMEGA) CHECKING FOR
C ERRORS AS EACH MEMBER OF THE SET IS READ IN.
DO 140 IPKP=1,KFCOMP

```

```

      READ(5,135) (OMEGA(IPKP,1),I=1,2)
135 FORMAT(214)
      TT=OMEGA(IPKP,1)
      IF(TT.LT.1) GO TO 145
      IF(TT.GT.KSTMLI) GO TO 145
      SS=OMEGA(IPKP,2)
      IF(SS.LT.1) GO TO 145
      IF(SS.GT.KSTMLI) GO TO 145
      GO TO 140
145 WRITE(6,150) IPKP,OMEGA(IPKP,1),OMEGA(IPKP,2)
150 FORMAT(1H1," THE",I4," FORCED PAIRED COMPARISON CONTAINS A STIMULU
-S THAT IS OUT OF RANGE CHECK ("I3," ",I3,"). *ERROR 8")
      KERROR=1
      IF(KERROR.EQ.1) GO TO 9999
140 CONTINUE
C
C   READ IN ALL THE VALUES FOR THE ATTRIBUTES ONE STIMULUS AT A TIME
C
      DO 155 KIT=1,KSTMLI
      READ(5,160) (STIMLI(KIT,JEST),JEST=1,KATTA)
160 FORMAT(8F10.0)
155 CONTINUE
      M=KFCOMP+1
C   DETERMINE THE COEFFICIENTS OF THE WEIGHTS IN THE CONSTRAINT MATRIX
C   BY SUBTRACTING THE SCORE OF A CERTAIN ATTRIBUTE W/R TO A STIMULUS I
C   FOR THE SCORE OF THE SAME ATTRIBUTE W/R TO A STIMULUS J WHEN J IS
C   PREFERRED TO I IN A FORCED PAIRED COMPARISON
      IF(IABS(KWFLAG)-1) 165,170,175
C   ALL WEIGHTS ARE EITHER NEGATIVE OR POSITIVE
170 DO 100 IABC=1,KFCOMP
      KFIRST=OMEGA(IABC,1)
      KSECND=OMEGA(IABC,2)
C   DETERMINE THE I' TH COEFFICIENT ROW OF THE CONSTRAINT MATRIX WHERE ROW I
C   CORRESPONDS TO THE I' TH FORCED PAIRED COMPARISON
      DO 100 JDEF=1,NUMWTS
      RHALF(IABC,JDEF)=KWFLAG*(STIMLI(KFIRST,JDEF)-STIMLI(KSECND,JDEF))
C   KEEP A RUNNING TOTAL TO DETERMINE THE M' TH ROW OF THE CONSTRAINT MATRIX
      RHALF(M,JDEF)=RHALF(M,JDEF)+RHALF(IABC,JDEF)
100 CONTINUE
      GO TO 210
C   ALL WEIGHTS ARE UNRESTRICTED
175 NUMWTS=2*NUMWTS
      DO 200 IABC=1,KFCOMP
      KFIRST=OMEGA(IABC,1)
      KSECND=OMEGA(IABC,2)
C   DETERMINE THE COEFFICIENT OF THE CONSTRAINT MATRIX WHEN ALL THE WEIGHTS ARE
C   UNRESTRICTED
      DO 200 JDEF=1,NUMWTS,2
C   COEFFICIENT OF WEIGHT(JDEF)+ AND CONSTRAINT IABC
      KSCT=(JDEF+1)/2
      RHALF(IABC,JDEF)=STIMLI(KFIRST,KSCT)-STIMLI(KSECND,KSCT)
C   COEFFICIENT OF WEIGHT(JDEF)- AND CONSTRAINT IABC
      JDEF2=JDEF+1

```

```

      RHALF(IABC,JDEF2)=-RHALF(IAEC,JDEF)
C      A RUNNING TOTAL OF THE M'TH ROW OF THE CONSTRAINT MATRIX
      RHALF(M,JDEF)=RHALF(M,JDEF)+RHALF(IAEC,JDEF)
      RHALF(M,JDEF2)=-RHALF(M,JDEF)
200 CONTINUE
      GO TO 210
C      THE WEIGHTS ARE MIXED
165 DO 300 IABC=1,KFCOMP
      KFIRST=OMEGA(IAEC,1)
      KSECND=OMEGA(IAEC,2)
      KTEMP=0
C      KTEMP IS A RUNNING TOTAL OF THE NUMBER OF COLUMNS TAKEN UP BY THE WEIGHTS
      DO 300 JDEF=1,NUMWTS
      IF(IABS(KW(JDEF)).GT.1) GO TO 165
C      THE JDEF'TH WEIGHT IS EITHER NEGATIVE OR POSITIVE
      KTEMP=KTEMP+1
      RHALF(IABC,KTEMP)=KW(JDEF)*(STIMLI(KFIRST,JDEF)-STIMLI(KSECND,
-JDEF))
      RHALF(M,KTEMP)=RHALF(M,KTEMP)+RHALF(IAEC,KTEMP)
      GO TO 300
C      THE JDEF'TH WEIGHT IS UNRESTRICTED
165 KTEMP=KTEMP+1
      RHALF(IABC,KTEMP)=STIMLI(KFIRST,JDEF)-STIMLI(KSECND,JDEF)
      RHALF(M,KTEMP)=RHALF(M,KTEMP)+RHALF(IAEC,KTEMP)
      ISST=KTEMP
      KTEMP=KTEMP+1
      RHALF(IABC,KTEMP)=-RHALF(IAEC,ISST)
      RHALF(M,KTEMP)=-RHALF(M,ISST)
300 CONTINUE
      NUMWTS=KTEMP
210 CONTINUE
C      CHECK TO SEE IF THE M'TH ROW OF RHALF IS ZERO, IS SO THEN THIS MEANS
C      THAT THE M'TH CONSTRAINT CANNOT BE SATISFIED THEREFORE THE PROBLEM
C      IS INFEASIBLE
      DO 1234 ICHCK=1,NUMWTS
      IF(RHALF(M,ICHCK).NE.0) GO TO 1235
1234 CONTINUE
      WRITE(6,1212)
1212 FORMAT(1H ,//," THE LEFT HAND SIDE OF THE M'TH CONSTRAINT IS "
-"ZERO.",//," THE STRUCTURE OF THE DATA SET CAUSES THE PROBLEM"
-" TO BE INFEASIBLE")
      GO TO 9999
1235 CONTINUE
C      CHECK TO SEE IF THE M'TH ROW OF RHALF CAN SATISFY THE
C      RESTRICTION ON THE WEIGHTS, IF ONE EXIST
      IF(KWFLAG.EQ.2) GO TO 1999
      DO 2111 ICHCK=1,KATTA
      IF(KW(ICHCK).EQ.2) GO TO 1999
      IF(RHALF(M,ICHCK).GT.0) GO TO 1999
2111 CONTINUE
C      THE PROBLEM IS INFEASIBLE, IT CANNOT SATISFY THE M'TH CONSTRAINT
      WRITE(6,6686)
6686 FORMAT(1H ,//," BECAUSE OF THIS PARTICULAR SPECIFICATION"

```

- " OF WEIGHT RESTRICTIONS", /, " THE PROBLEM IS INFEASIBLE"
 -, " LETTING ALL WEIGHTS BE UNRESTRICTED WILL WORK")
 GO TO 9999

1999 CONTINUE

C SEND ALL THE DEFINED COLUMNS OF THE FIRST NUMWTS COLUMNS OF RHALF TO AMP

C

CALL XMAPS(BNDTIP, IOERR, LENMA, LENMI, LENMY, MAPA, MAPI, MAXA, MAXM
 -, IAXN, MEMORY)

C THE FIRST M-1 CONSTRAINTS ARE ≥ 0 WITH A RHS VALUE OF 0.

DO 400 II=1, KFCOMP

ROWTYP(II)=-1

B(II)=0.

400 CONTINUE

C THE M'TH CONSTRAINT IS =1

ROWTYP(M)=0

B(M)=1.

N=0

C

C DETERMINE THE NUMBER OF NONZERO ELEMENTS(COLLEN) IN THE FIRST NUMWTS
 C COLUMNS OF THE CONSTRAINT MATRIX. ALSO DETERMINE WHICH ROWS(I.E.COLI) HAVE
 C NONZERO ELEMENTS IN THEM FOR A CERTAIN COLUMN AND WHAT THAT NONZERO
 C ELEMENT IS(I.E. COLA)

CJ=0.

C CJ IS THE OBJECTIVE FUNCTION COEFFICIENT

DO 600 JUST=1, NUMWTS

COLLEN=0

DO 500 LMN=1, M

IF(RHALF(LMN, JUST).EQ.0) GO TO 500

COLLEN=COLLEN+1

COLA(COLLEN)=RHALF(LMN, JUST)

COLI(COLLEN)=LMN

500 CONTINUE

C CHECK FOR TO SEE IF THIS COLUMN HAS ALL ZEROS IN IT

C ALSO FLAG THE UNDEFINED WEIGHTS

IF(COLLEN.EQ.0) KNDDEF(JUST)=1

IF(COLLEN.EQ.0) GO TO 600

C PASS THIS COLUMN TO AMP

C

CALL XADDAJ(CJ, COLA, COLI, COLLEN, COLMAX, IOERR, J, LENMA, LENMY, MAPA,
 -, MEMORY, N)

C

600 CONTINUE

C CHECK TO SEE IF THERE EXIST ANY WEIGHTS THAT ARE DEFINED

IF(N.NE.0) GO TO 602

WRITE(6, 601)

601 FORMAT(1H, "ALL WEIGHTS ARE UNDEFINED WHICH MAKES IT IMPOSSIBLE"

-, " TO SOLVE THIS PARTICULAR PROBLEM WITH THIS DATA SET")

GO TO 9999

602 CONTINUE

C SEND THE REMAINING COLUMNS(I.E. THE 2'S) TO AMP

C NOTE: EACH 2 COLUMN HAS ONE NON-ZERO ELEMENT AND AN OBJECTIVE FUNCTION

C COEFFICIENT OF 1, BUT SINCE AMP ONLY MAXIMIZES WE MUST USE -1 AS THE

C OBJECTIVE FUNCTION COEFFICIENT

```

COLLEN=1
CJ=-1.
COLA(1)=1.
DO 700 J2=1,KFCOMP
COLI(1)=J2
-----
CALL XADDAJ(CJ,COLA,COLI,COLLEN,COLMAX,IOERR,J,LENMA,LENNY,
-MAPA,MEMORY,N)
-----
700 CONTINUE
-----
DO 710 JXJ=1,N
STATUS(JXJ)=0
710 CONTINUE
-----
CALL XSLACK(B,BASCB,BASIS,EASLB,EASUB,BNDTYP,BOUND,COLA,COLI,
X COLMAX,IOERR,LENMA,LENNI,LENNY,N,MAPA,MAPI,MAXN,MAXN,MEMORY,N,
X ROWTYP,STATUS,UZERO,XBZERO,Z)
-----
SOLVE THE L.P. PROBLEM
-----
CALL XPINIL(B,BASCB,BASIS,EASLB,EASUB,BNDTYP,BOUND,CAND,CANDA,
X CANDCJ,CANDI,CANCL,COLA,COLI,COLMAX,FACTOR,IOERR,ILOG,ITER1,
X ITER2,LENMA,LENNI,LENNY,LOCK,N,MAPA,MAPI,MAXN,MAXN,MEMORY,N,
X NTYPE2,P,PRINT,STATUS,TERMIN,ONBDLQ,UZERO,XBZERO,YQ,Z)
-----
WRITE(6,742) (HEADER(IHEAD),IHEAD=1,10)
742 FORMAT(1H1,/,1X,10A5,////)
WRITE OUT THE TERMINATION CODE OF THE L.P. PROBLEM TO TAPE 9
WRITE(9,720) TERMIN
720 FORMAT(1H,"THE TERMINATION CODE IS",I5)
IF(TERMIN.EQ.1) GO TO 740
IF(TERMIN.NE.4) GO TO 9999
WRITE(6,1358)
1358 FORMAT(1H,"***WARNING***",10X,"***WARNING***",10X,"***WARNING***"
-,10X,"***WARNING***",/," THE OPTIMAL SOLUTION DOES NOT SATISFY",
-" THE ACCURACY CHECK FROM XMP,TERMINATION CODE 4.")
-----
TERMIN IS THE TERMINATION CODE AND IS EXPLAINED IN THE DOCUMENTATION
OF XMP
-----
DETERMINE THE VALUES OF THE WEIGHTS FROM THE XMP VARIABLES
740 IF(IABS(KWFLAG)-1) 725,730,735
.....
KWT KEEPS A RUNNING TOTAL OF WHAT XMP VARIABLE YOU ARE TALKING ABOUT
WITH/RESPECT TO THE ARRAY STATUS.
730 KWT=0
ALL WEIGHT ARE EITHER NEGATIVE OR POSITIVE
DO 800 IJJ=1,NUMWTS
CHECK TO SEE IF THE IJJ'TH WEIGHT IS UNDEFINED

```

```

      IF(KWDEF(IJJ).EQ.1) GO TO 800
      KWT=KWT+1
C     CHECK TO SEE IF THE IJJ VARIABLE IS IN THE BASIS
      IF(STATUS(KWT).EQ.0) GO TO 745
      WEIGHT(IJJ)=FLOAT(KWFLAG)*AEZERO(STATUS(KWT))
      GO TO 800
745  WEIGHT(IJJ)=0.
800  CONTINUE
      GO TO 801
C     .....
C     ALL WEIGHTS ARE UNRESTRICTED
735  KOUNT=1
      KWT=0
      DO 1000 IJJ=1,NUMWTS,2
C     CHECK TO SEE IF THE KOUNT'TH WEIGHT IS UNDEFINED
      IF(KWDEF(IJJ).EQ.1) KWDEF(KOUNT)=1
      IF(KWDEF(IJJ).EQ.1) GO TO 757
      KWT=KWT+1
C     THE KOUNT'TH WEIGHT IS DEFINED
      KWDEF(KOUNT)=0
C     DETERMINE THE KOUNT'TH WEIGHT IF IT IS DEFINED
      LIK=KWT
765  IF(STATUS(KWT).EQ.0) GO TO 747
C     THE KOUNT'TH WEIGHT IS NONZERO
      WEIGHT(KWT)=AEZERO(STATUS(KWT))
      GO TO 750
747  WEIGHT(KWT)=0.
750  CONTINUE
C     CHECK TO SEE IF WE HAVE ALREADY CHECKED BOTH THE POSITIVE AND NEGATIVE
C     PARTS OF THE KOUNT'TH WEIGHT
      IF(LIK.NE.KWT) GO TO 755
      KWT=KWT+1
C     WHEN LIK=KWT WE ARE CHECKING TO SEE IF THE NONNEGATIVE PART OF THE
C     KOUNT'TH WEIGHT IS POSITIVE, SIMILARLY IF KWT=LIK+1 WE ARE CHECKING TO
C     SEE IF THE NONPOSITIVE PART OF THE KOUNT'TH WEIGHT IS NEGATIVE
      GO TO 765
755  WEIGHT(KOUNT)=WEIGHT(LIK)-WEIGHT(KWT)
757  KOUNT=KOUNT+1
1000 CONTINUE
      GO TO 801
C     .....
C     THE WEIGHTS ARE MIXED
725  KTEMP=1
      KFLAG=0
C     KTEMP KEEPS A RUNNING TOTAL ON THE NUMBER OF WEIGHTS, ACTUAL WEIGHTS AND
C     FOR EXAMPLE IF THERE ARE 10 WEIGHTS >= 0 AND 5 UNRESTRICTED WEIGHTS THEN
C     THE FINAL VALUE OF KTEMP WILL BE 15.
      KWT=0
      DO 1100 MNO=1,NUMWTS
      IF(KFLAG.NE.1) GO TO 1730
      KFLAG=0
      GO TO 1100
1730 IF(ABS(KW(KTEMP)).EQ.1) GO TO 1745

```

```

C   THE KTEMP'TH WEIGHT IS UNRESTRICTED
    KFLAG=1
C   CHECK TO SEE IF THE KOUNT'TH WEIGHT IS UNDEFINED
    IF(KWDEF(MNO).EQ.1) KWDEF(KOUNT)=1
    IF(KWDEF(MNO).EQ.1) GO TO 1770
    KWT=KWT+1
C   THE KOUNT'TH WEIGHT IS DEFINED
    KWDEF(KOUNT)=0
    IF(STATUS(KWT).EQ.0) GO TO 1755
    WEIGHT(KTEMP)=XBZ=KO(STATUS(KWT))
    KWT=KWT+1
    GO TO 1770
1755 KWT=KWT+1
    IF(STATUS(KWT).NE.0) GO TO 1766
    WEIGHT(KTEMP)=0.
    GO TO 1770
1766 WEIGHT(KTEMP)=-XBZERO(STATUS(KWT))
    GO TO 1770
1745 CONTINUE
C   THE KTEMP'TH WEIGHT IS EITHER - OR + ,BUT IS NOT UNRESTRICTED
C   CHECK TO SEE IF THE KOUNT'TH WEIGHT IS UNDEFINED
    IF(KWDEF(MNO).EQ.1) KWDEF(KOUNT)=1
    IF(KWDEF(MNO).EQ.1) GO TO 1770
    KWT=KWT+1
    IF(STATUS(KWT).NE.0) GO TO 1775
    WEIGHT(KTEMP)=0.
    GO TO 1770
1775 WEIGHT(KTEMP)=KW(KTEMP)*XBZERO(STATUS(KWT))
1770 KTEMP=KTEMP+1
1100 CONTINUE
C   .....
801 CONTINUE
C   DETERMINE THE VALUE OF THE Z'S FOR ALL KWFLAGS
    KSCRPT=NUMPTS
    DO 22 IJK=1,KATTA
    IF(KWDEF(IJK).EQ.1) KSCRPT=KSCRPT-(IABS(KW(IJK)))
22 CONTINUE
    DO 900 IIJ=1,KFCORP
    KSCRPT=KSCRPT+1
    IF(STATUS(KSCRPT).EQ.0) GO TO 815
    Z2(IIJ)=XBZERO(STATUS(KSCRPT))
    GO TO 900
815 Z2(IIJ)=0.
900 CONTINUE
C   SINCE WE USED XMP TO MAXIMIZE AND IN REALITY WE WANTED TO MINIMIZE WE HAD
C   TO MULTIPLY THE OBJECTIVE FUNCTION COEFFICIENTS BY -1 SO OUR Z(OPTIMAL)
C   OBJECTIVE FUNCTION VALUE) IS REALLY --2
    Z=-2
    WRITE(6,1920) Z
1920 FORMAT(1H , "THE OPTIMAL OBJECTIVE FUNCTION VALUE IS Z=",F10.5)
C   WRITE OUT THE HEADING FOR THE OUTPUT
    WRITE(6,1930)
1930 FORMAT(1H , "//,10X,"WEIGHT#",17X,"***VALUE**")

```

```

        WRITE(6,1935)
1935  FORMAT(1H ,9X,"-----",17X,"-----")
C      WRITE OUT THE VALUE OF THE WEIGHTS
C      IDEFLG IS A FLAG WHICH IS EQUAL TO ONE IF THERE ARE ANY UNDEFINED WEIGHTS
        IDEFLG=0
        DO 1940 IJJ=1,KATTA
C      CHECK TO SEE IF THE KATTA' TH WEIGHT IS UNDEFINED
        IF(KWDEF(IJJ).EQ.0) GO TO 1936
        WRITE(6,1942) IJJ
1942  FORMAT(12X,I3,16X,"UNDEFINED")
        IDEFLG=1
        GO TO 1940
1936  WRITE(6,1950) IJJ,WEIGHT(IJJ)
1950  FORMAT(12X,I3,17X,F10.5)
1940  CONTINUE
        WRITE(6,1960)
1960  FORMAT(1H1,////,14X,"2",19X,"***VALUE**")
        WRITE(6,1961)
1961  FORMAT(1H ,12X,"---",16X,"-----")
C      WRITE OUT THE VALUES OF THE Z'S
        DO 1200 IJKLM=1,KFCOMP
        WRITE(6,1185) IJKLM,22(IJKLM)
1185  FORMAT(1H ,11X,I3,17X,F10.5)
1200  CONTINUE
C      CALCULATE THE SCORES FOR EACH STIMULUS AND WRITE THEM OUT
        WRITE(6,2022)
2022  FORMAT(1H1,////,9X,"STIMULUS",17X,"***SCORE**")
        WRITE(6,2023)
2023  FORMAT(1H ,8X,"-----",17X,"-----")
        DO 1202 KIT=1,ASTMLI
        SCORE=0.
        DO 1203 KRR=1,KATTA
        IF(KWDEF(KRR).EQ.1) GO TO 1203
        SCORE=SCORE+WEIGHT(KRR)*STIMLI(KIT,KRR)
1203  CONTINUE
        WRITE(6,2229) KIT,SCORE
2229  FORMAT(1H ,11X,I3,17X,F10.5)
1202  CONTINUE
C      WRITE OUT A MESSAGE IF ANY OF THE WEIGHTS ARE UNDEFINED, I.E. IDEFLG=1
        IF(IDEFLG.EQ.0) GO TO 1204
        WRITE(6,17)
17  FORMAT(1H ,//," FOR THE DATA SET USED IN THIS PROBLEM SOME OF THE"
        -," ABOVE WEIGHTS COULD NOT BE DETERMINED")
1204  WRITE(6,1976)
1976  FORMAT(1H1,///," PROGRAM TERMINATES NORMALLY")
9999  CONTINUE
        STOP
        END
        END

```

**LATE
LME**